

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Canceled)
2. (Canceled)
3. (Previously Presented): A front end compiler system for generating code to be used by an execution environment, said front end system comprising:
 - a metadata module that compiles information to produce metadata information;
 - a code module that compiles information to produces executable instructions; and
 - wherein the metadata information and executable instructions are the result of compiling a source file in a first language and the front end compiler consumes metadata information produced by a different front end compiler as a result of compiling a source file in a second language.
4. (Canceled)
5. (Canceled)
6. (Canceled)
7. (Previously Canceled)
8. (Previously Canceled)
9. (Previously Canceled)
10. (Previously Canceled)
11. (Previously Canceled)
12. (Previously Canceled)
13. (Previously Canceled)
14. (Previously Canceled)
15. (Previously Canceled)

16. (Canceled)

17. (Original) A method of compiling a computer program written in a native source language and having an import statement that imports a common language file, said method comprising:

parsing the computer program;

examining each statement during the parsing act and determining if the statement is an import statement related to the common language file;

if the statement relates to the common language file, reading the common language file into a symbol table;

if the statement relates to a native language symbol table entry, adding the information into the symbol table; and

if the statement relates to output generation, supplying the statement to a output generator.

18. (Presently Amended) A method as defined in claim 17 wherein the common language file ~~may be~~ is imported by different source language files.

19. (Presently Amended) A method as defined in claim 17 wherein the common language file ~~may be~~ is imported by a procedural source language file and an object oriented source language file.

20. (Presently Amended) A method as defined in claim 17 wherein the computer program is written in a procedural programming language and the common language file ~~may be~~ is imported by an object oriented source language files.

21. (Original) A method as defined in claim 17 wherein the output generator produces a second common language file wherein the second common language file is different from the imported common language file.

22. (Original) A method as defined in claim 21 wherein the second common language file has a metadata section and a common language instructions section.

23. (Original) A method as defined in claim 22 wherein the act of reading the common library file into a symbol table further comprises reading the metadata into the symbol table.

24. (Canceled)

25. (Original) A computer readable medium having stored thereon a data structure comprising a common language file produced by a front end compiler that consumes a native source code file written in a native source code language, wherein the native source code language is one of a plurality of source languages, the common language file comprising:

a common language instructions section having instructions in a common language, the instructions related to the written program functions of the native source code file and consumed metadata, wherein the metadata describes written program functions of another native source code language file, the common language used to represent the written program functions is adapted to represent written program functions originally written in at least two different source code languages;

a metadata portion that describes the common language instructions in the common language instructions section; and

wherein the front end compiler is adapted to compile native source code files and another common language file.

26. (Presently Amended) A computer readable medium having stored thereon a data structure comprising a common language file produced by a front end compiler that consumes a native source code file written in a native source code language, wherein the native source code language is one of a plurality of source languages, the common language file comprising:

a common language instructions section having instructions in a common language, the instructions related to the written program functions of the native source code file and consumed metadata, wherein the metadata describes written program functions of another native source code language file, the common language used to represent the written program functions is adapted to represent written program functions originally written in at least two different source code languages;

Application No. 09/598,105

a metadata portion that describes the common language instructions in the common language instructions section; and

wherein the compiler is adapted to compile a native source code file, the native source code file utilizing a common language library, and wherein the common language library ~~may be~~ is consumed by different front end compilers associated with different native source code languages.